PUC Minas

DIRETORIA DE
EDUCAÇÃO CONTINUADA

Pós Graduação *Lato Sensu*

Bancos de dados não relacionais

Índices e Arquitetura

Armazenamento dos dados

Documento JSON

JSON

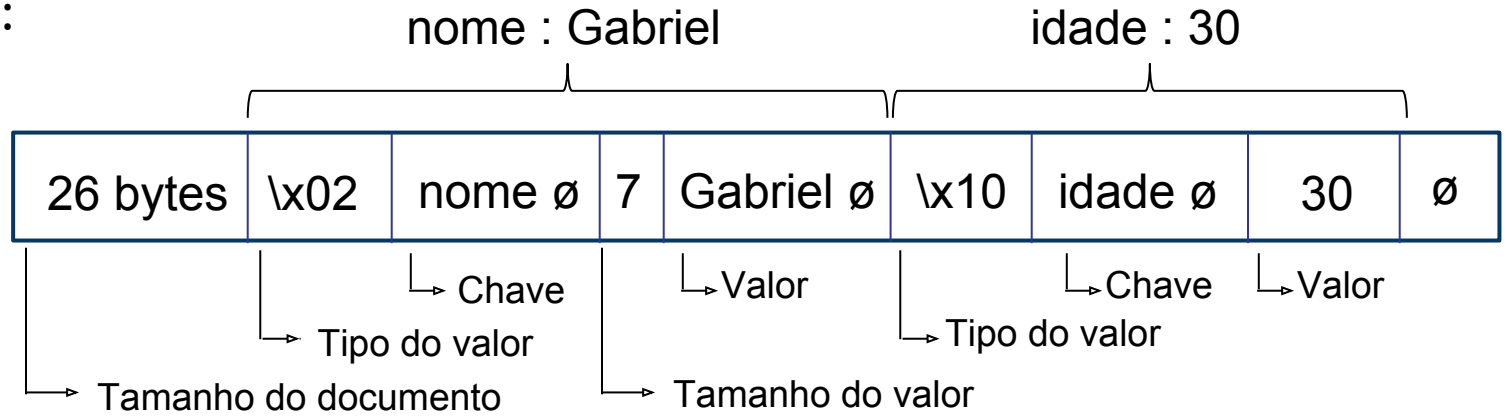
```
{
      “nome” : “Gabriel”,
      “idade”: 30
}
```

Documento BSON

JSON:

```
{
      “nome” : “Gabriel”,
      “idade”: 30
}
```

BSON:



Document JSON -> BSON

- JSON é ~~mais~~ legível

Porém o BSON:

- Mais compacto
- Mais fácil e rápido de transportá-lo
- Mais fácil e mais rápido de escaneá-los

Eficiência

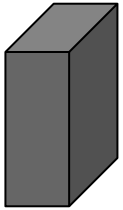
Armazenamento em NoSQL

- Não possui **transações** para múltiplos documentos;
- Controle de **concorrência** a nível de documento para escrita;
- **Múltiplos** clientes podem modificar diferentes documentos ao **mesmo** tempo;
- **Conflito** entre duas operações uma delas “levanta” o erro.

Resolução de conflitos

Usuário

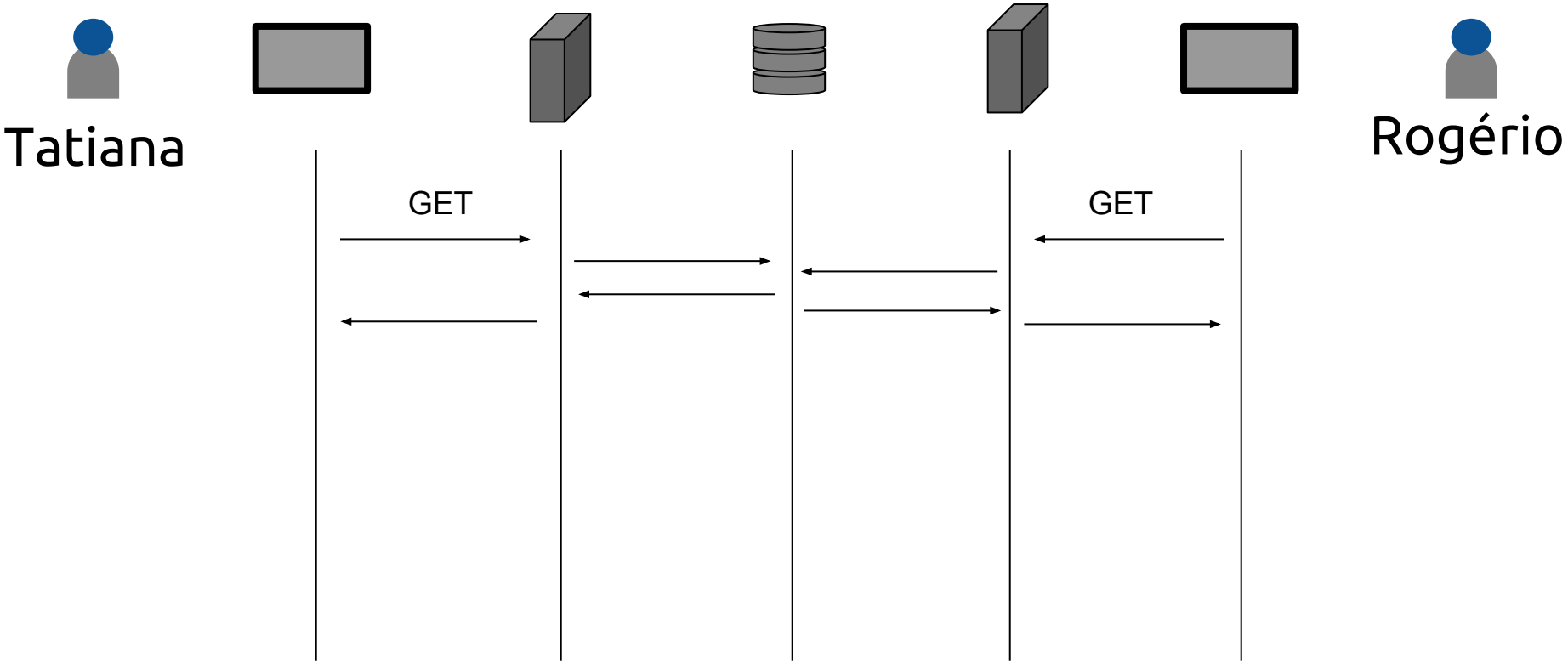
Browser



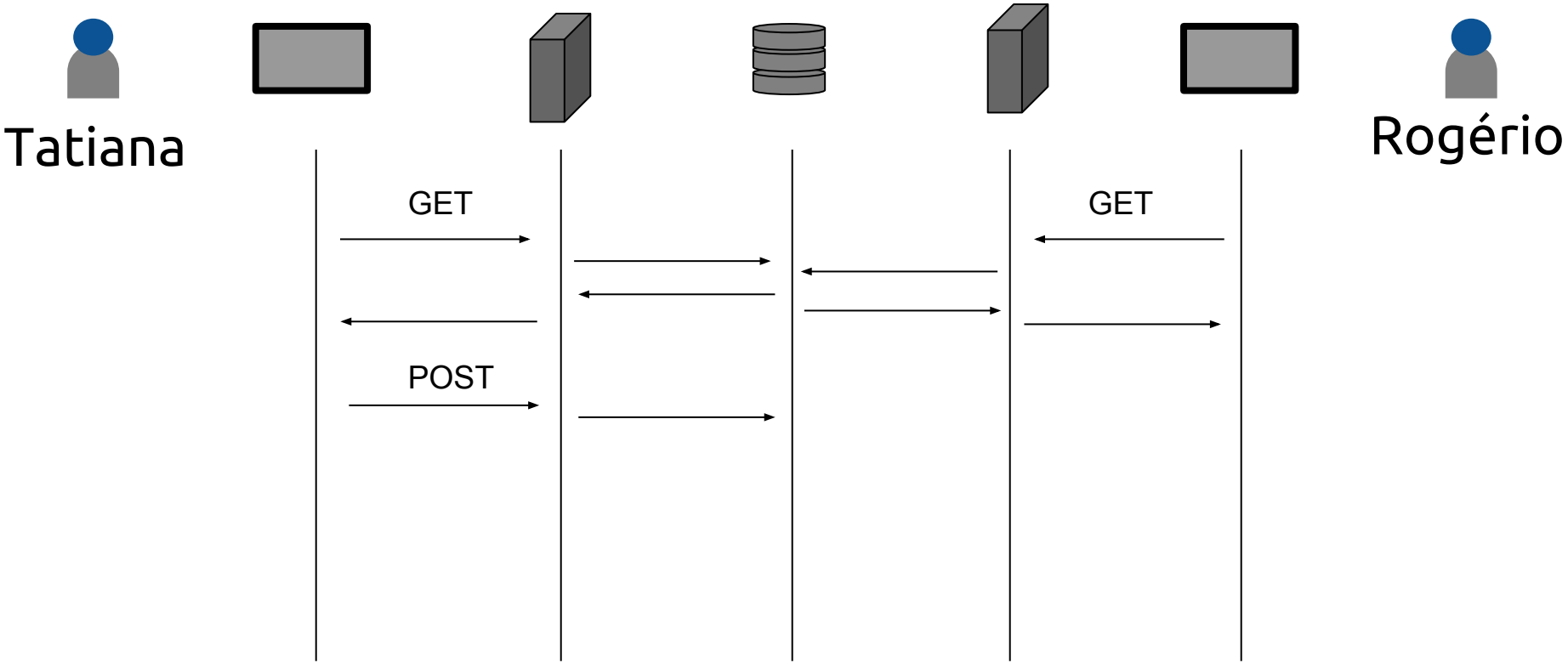
Servidor

BD

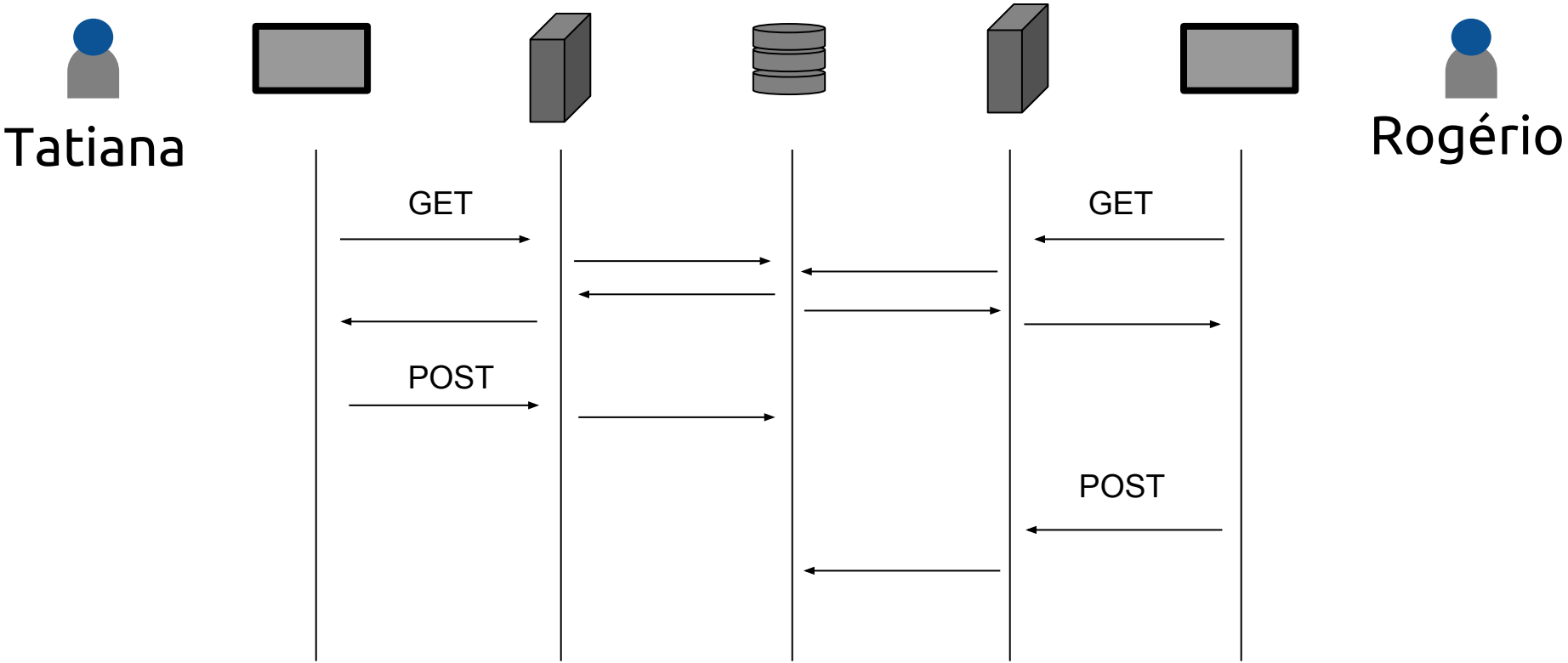
Resolução de conflitos



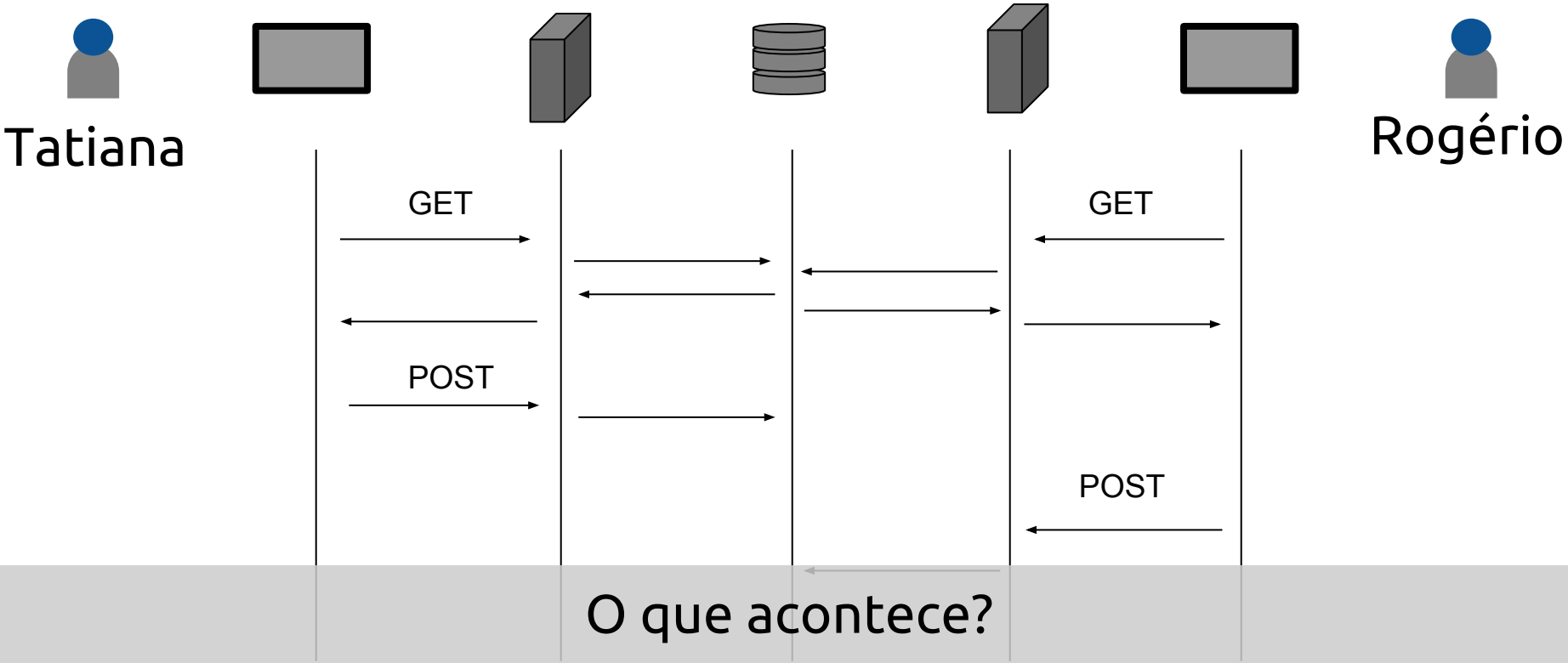
Resolução de conflitos



Resolução de conflitos

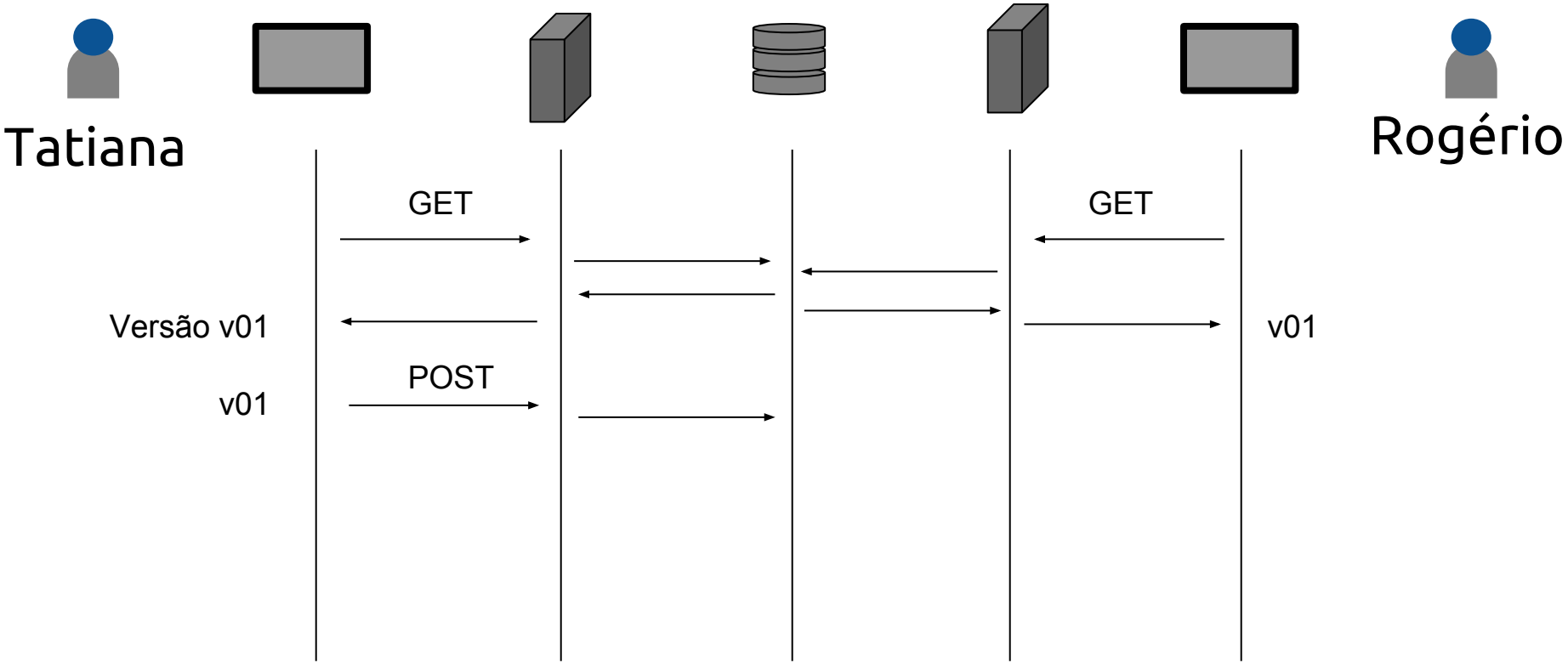


Resolução de conflitos

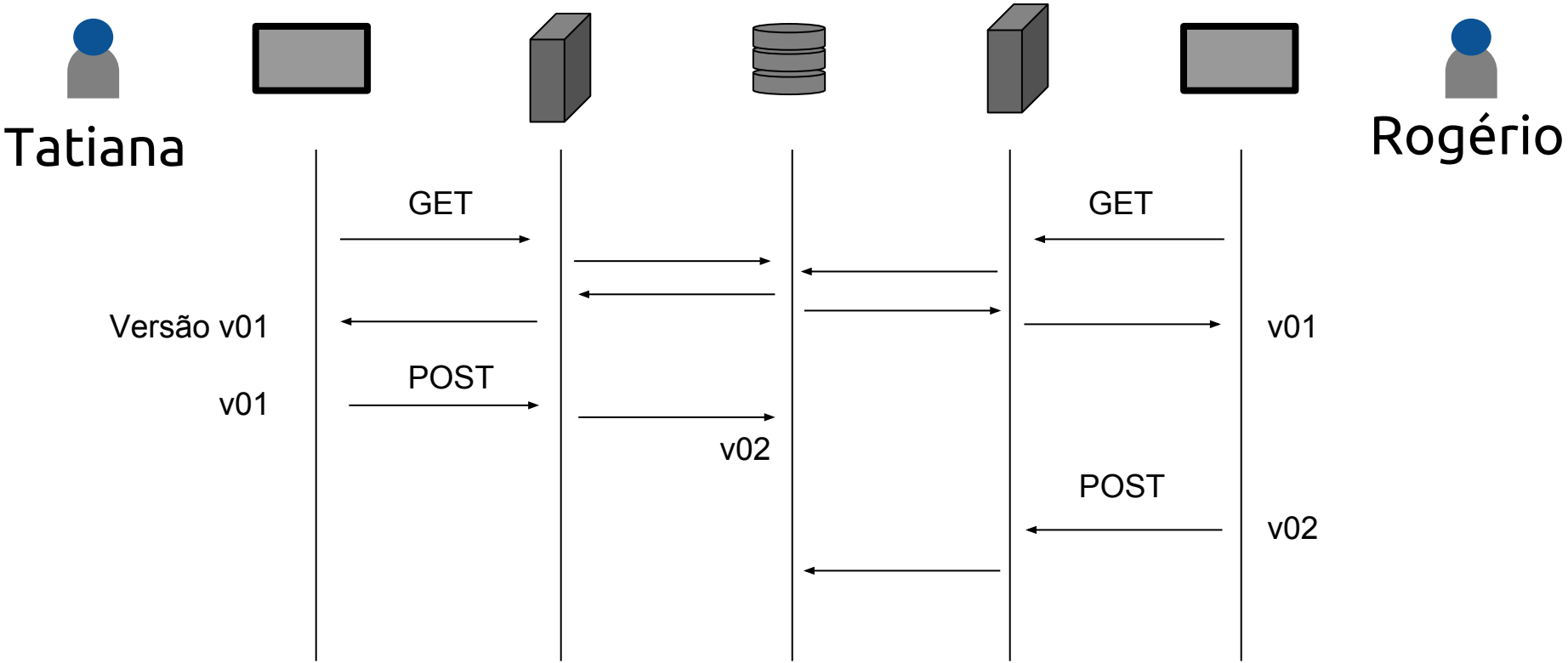


O que acontece?

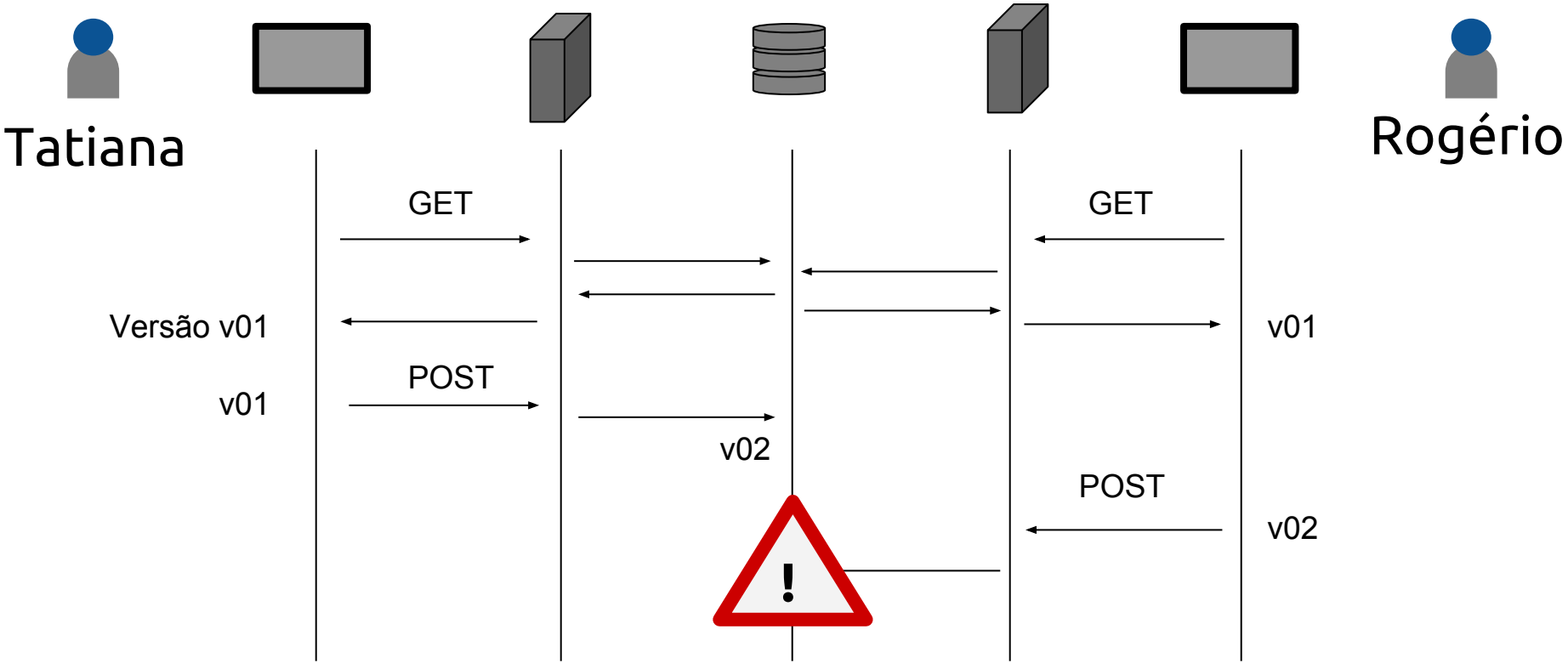
Resolução de conflitos



Resolução de conflitos

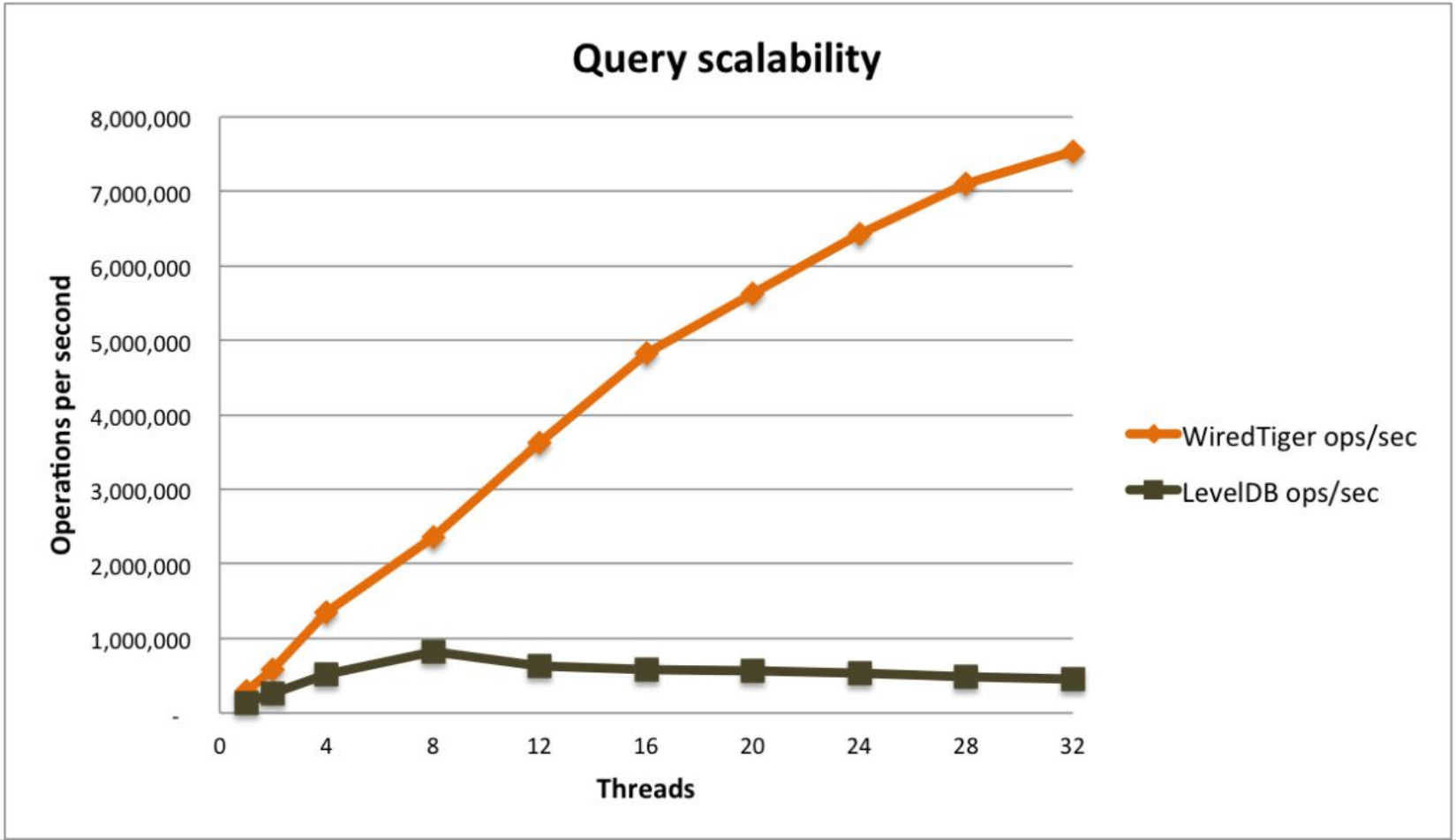


Resolução de conflitos

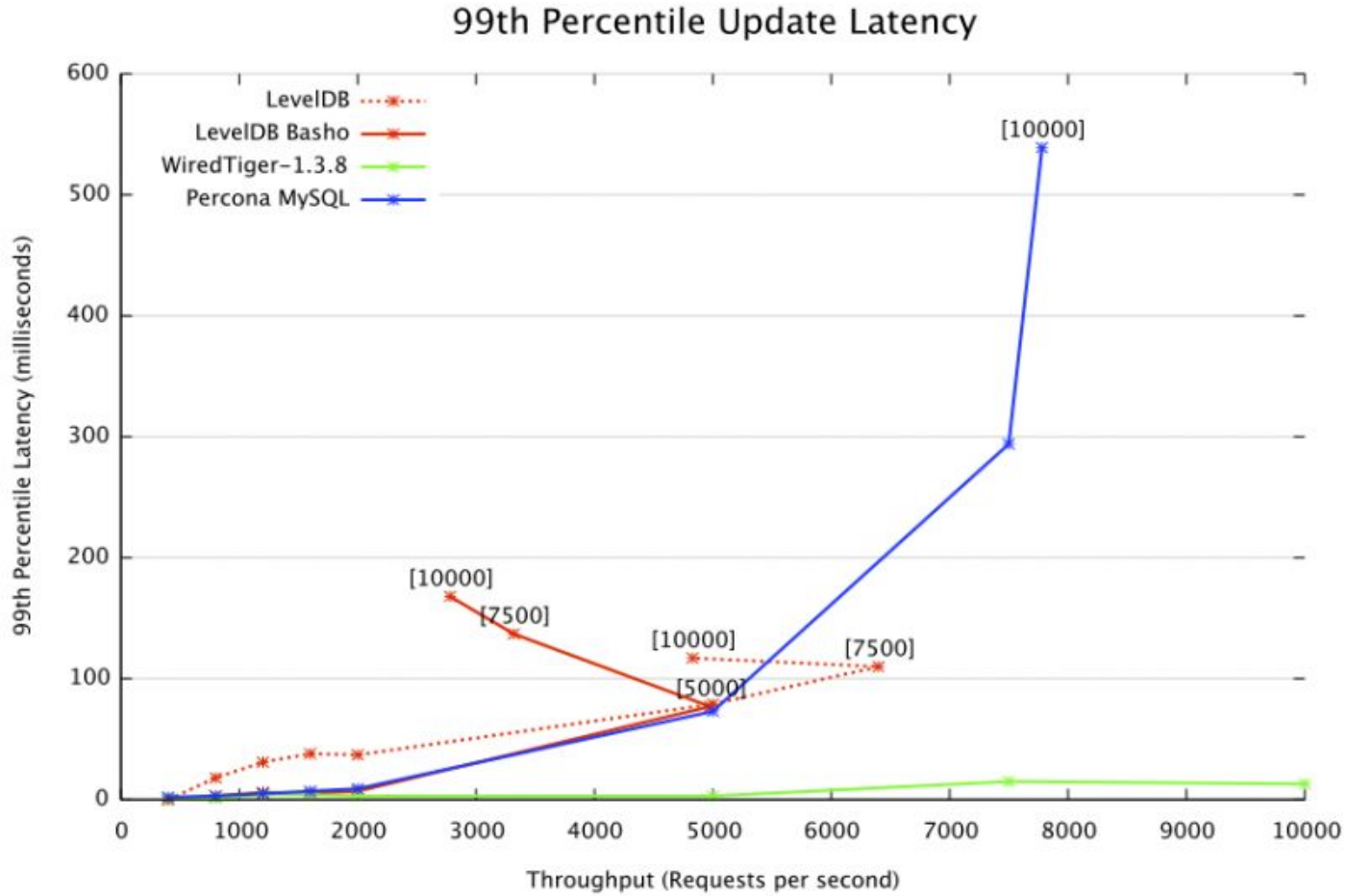


Journaling

- Histórico de ações
- *Checkpoints* para garantir a recuperação de falhas em caso de uma parada inesperada;
- **Sincronização** de dados de tempos em tempos em disco;
- Fornece **durabilidade** das operações em caso de falha.


Query scalability
Operations per second
8,000,000
7,000,000
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
-
0
4
8
12
16
20
24
28
32
Threads
WiredTiger ops/sec
LevelDB ops/sec


Insert Rate for iiBench
Inserts Per Second
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1
15
29
43
57
71
85
99
113
127
141
155
169
183
197
211
225
239
253
267
281
295
309
323
337
351
365
379
393
407
421
435
449
463
477
491
Row Count (millions)
WiredTiger
InnoDB


99th Percentile Update Latency
LevelDB
LevelDB Basho
WiredTiger-1.3.8
Percona MySQL
99th Percentile Latency (milliseconds)
Throughput (Requests per second)
0
100
200
300
400
500
600
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
[10000]
[7500]
[10000]
[5000]
[7500]
[10000]

GridFs - Armazenamento de arquivos

NoSQL é um **bom** lugar para armazenar arquivos desde que:

1. Você queira **associar** arquivos a metadados e você queira achar os arquivos por meio desses metadados;
2. **Simplificar** a arquitetura da sua aplicação;
3. Necessidade de uma aplicação **escalável**.

Índices

O que é?

Estrutura utilizada acelerar o tempo de acesso aos documentos.

Como funciona?

db.Alunos.count() // 10.000.000 documentos

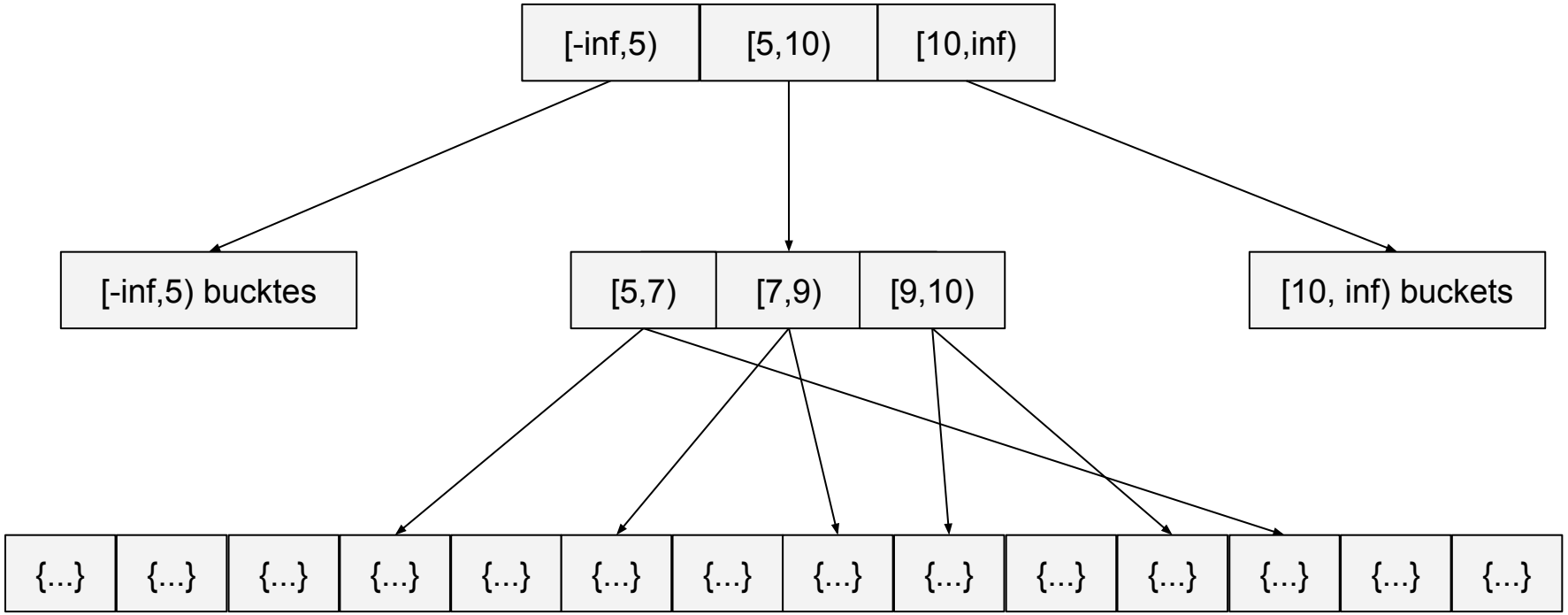
db.Alunos.find({“name”: /Gabriel/}) // 1s para retornar

Por que demora tanto tempo?

Como funciona?

- Diferentes estruturas de indexação
  - Btree
  - B+tree
  - Hashing
  - Invertido
  - etc

Como funciona? BTree



Como funciona?

Ao invez de percorrer 10.000.000

$$\text{Log}(10.000.000) = \sim 23$$

Tipos de índices

_id: Sempre indexado

Vantagens de se utilizar ObjectId:

ObjectId(“**4bface1a**223131**6e04**f3c434”)

Counter

ProcessId

Machine Id

Timestamp

Tipos de índices

```
{
    “name”: “Gabriel”,
    “score”: 18
}
```

Índice simples

```
db.user.createIndex({“nome”:1})
```

Índice composto

```
db.user.createIndex({“nome”:1,”score”:1})
```

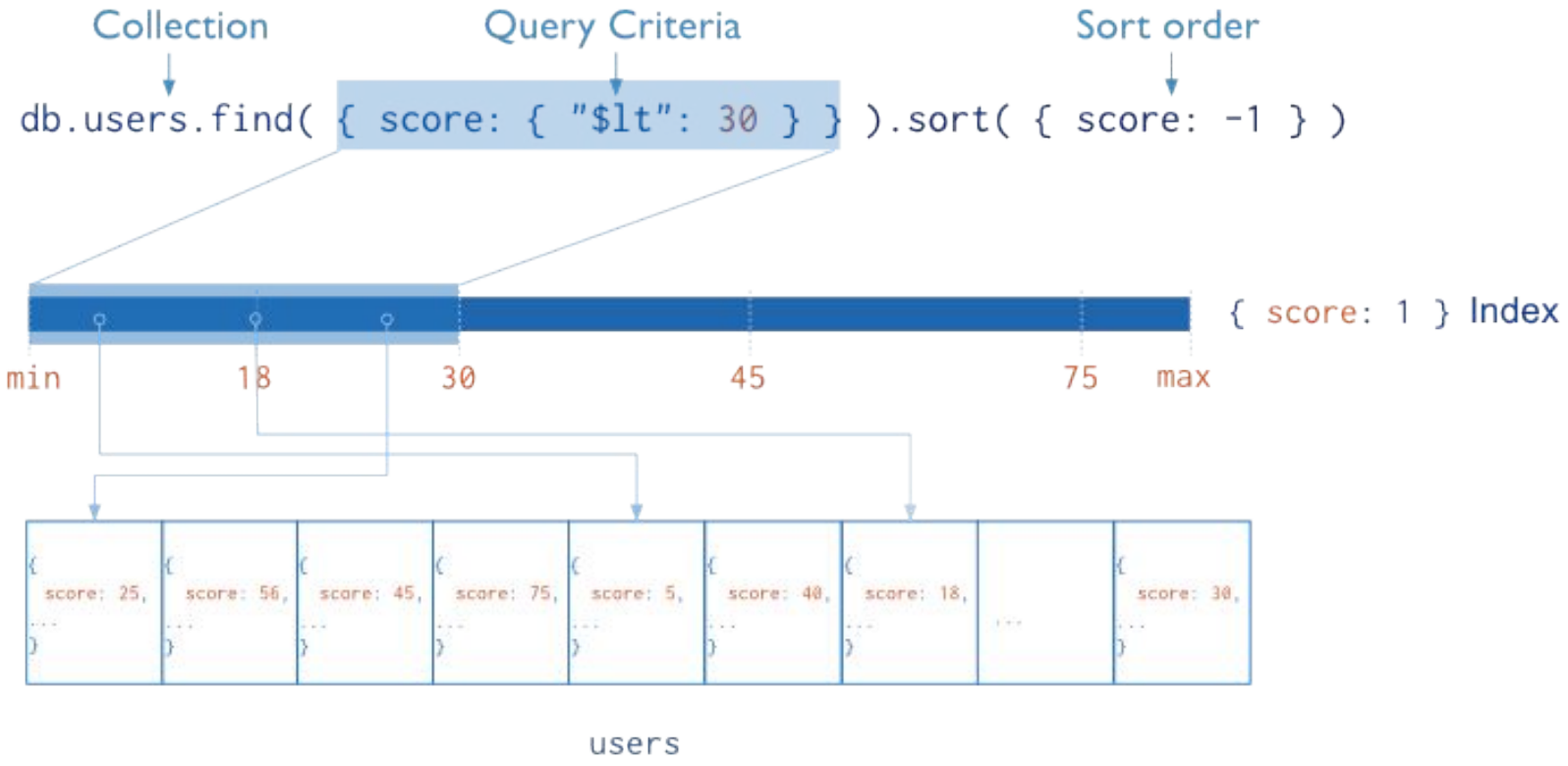
Atenção com índices compostos

Ao usar índice composto você pode:

db.user.find({“username”:”Gabriel”})
db.user.find({“username”:”Gabriel”}).sort({“score”:1})
db.user.find().sort({“score”:1,”nome”:1})

Note que fazendo um índice composto o índice simples em **username** não é mais necessário

Tipos de índices



Tipos de índices

Índice incorporado

```
{
      “name”: “Gabriel”,
      “information”: {
            “city”      : “Belo Horizonte”,
            “state” : “Minas Gerais”
      }
}
```

```
db.user.createIndex({“information”:1})
```

Tipos de índices

Índice único

db.user.createIndex({“name”:1},{“unique”:true})

Índice esparso ou parcial

db.user.createIndex({“score”:1}, {“sparse”:true})

db.user.createIndex({“score”:1}, {“partialFilterExpression”:true})

Tipos de índices

Índice textual

```
db.Aluno.createIndex({“nome”:”text”})
```

- Muito bom para textos com um número razoável de caracteres;
- Otimizações para armazenar informações;

Tipos de índices

Índice geoespacial

```
{
    “name”: ”place”
    “loc” : { type: "point", coordinates: [ -73.88, 40.78 ] }
}
```

```
db.places.createIndex( { loc : "2dsphere" } )
```

Tipos de índices - Capped Collections

Coleções com um tamanho máximo de documentos:

```
db.createCollection("log", { capped : true, size : 50, max : 50})
```

ou

```
db.runCommand({"convertToCapped": "mycoll", size: 10})
```

Tipos de índices - Capped Collections

Size: 3

{“name”:”test 1”}

Tipos de índices - Capped Collections

Size: 3

{“name”:”test 2”}

{“name”:”test 1”}

Tipos de índices - Capped Collections

Size: 3

{“name”:”test 3”}

{“name”:”test 2”}

{“name”:”test 1”}

Tipos de índices - Capped Collections

Size: 3

{“name”:”test 4”}

{“name”:”test 3”}

{“name”:”test 2”}

Tipos de índices - TTL

```
{
   “name”: “Aluno”
  “createdAt”: ISODate("2012-12-19T06:01:17.171Z")
}
```

```
db.log_events.createIndex( { "createdAt": 1 }, { expireAfterSeconds: 3600 } )
```

Atenção na criação de índices

- Crie índices que ajudem suas consultas;
- Use índices para facilitar a ordenação
- Garanta que os índices caibam na memoria RAM
- Crie índices que sejam seletivos

Depurando consultas

db.Aluno.find({"nm":"Gabriel Campos"}).explain("executionStats":1)

nReturned -> Número de objetos retornados
executionTimeMillis - > Tempo gasto em milisegundos
totalKeysExamined -> Total de chaves examinadas
totalDocsExamined -> Total de documentos examinadas

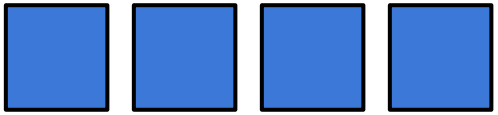
Exemplo prático

- Consulta sem índice
- Criar um indice
- Consulta com índice
- Exclusão de índice

Escalabilidade

Tipos de escalabilidade

Vertical

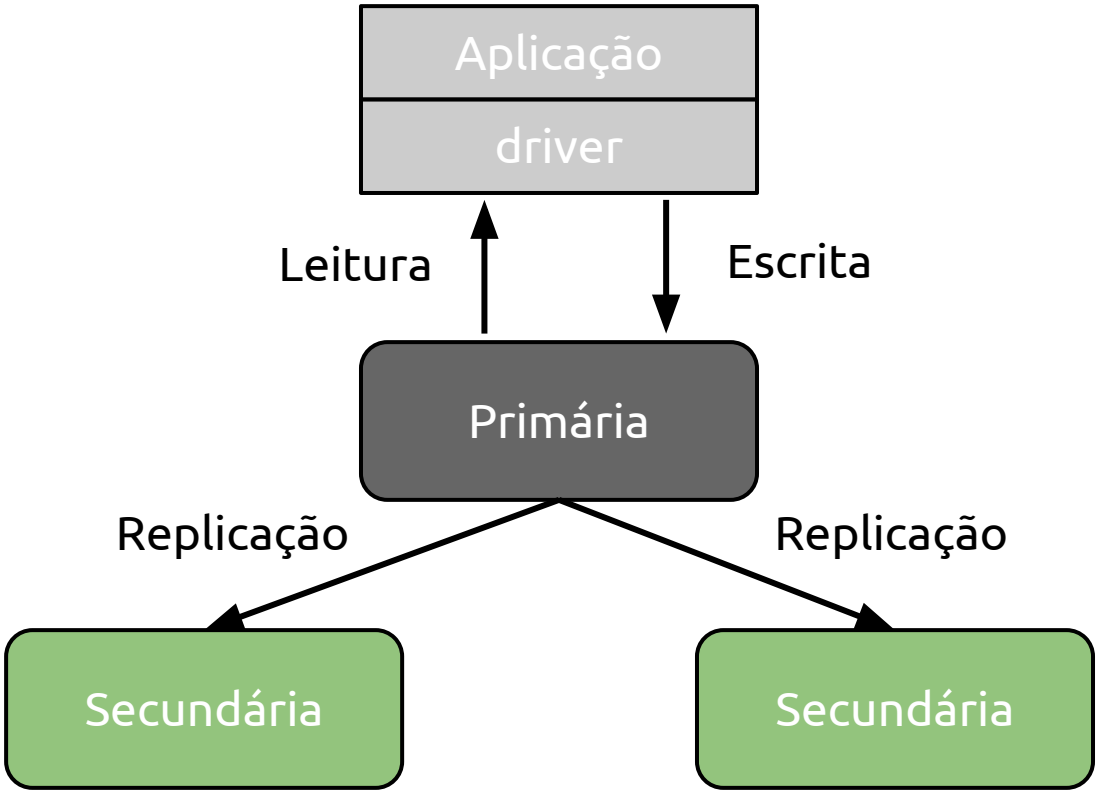


Horizontal

Replica Set

- Oferecer redundância e aumentar a **disponibilidade** dos dados
- Distribuindo em servidores **diferentes**
- Em caso de **falha** no servidor principal outro servidor assume o seu papel.

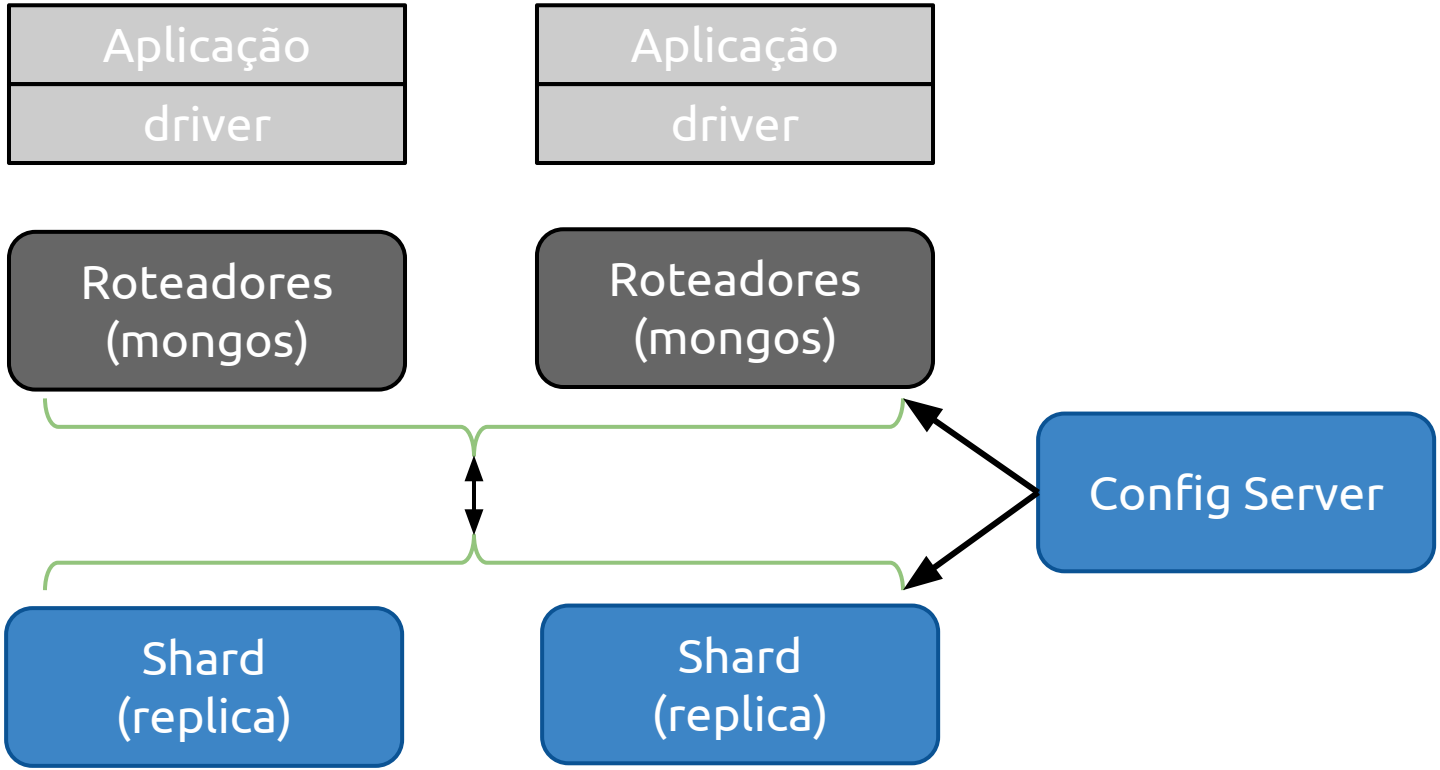
Replica Set


Aplicação
driver
Leitura
Escrita
Primária
Replicação
Replicação
Secundária
Secundária

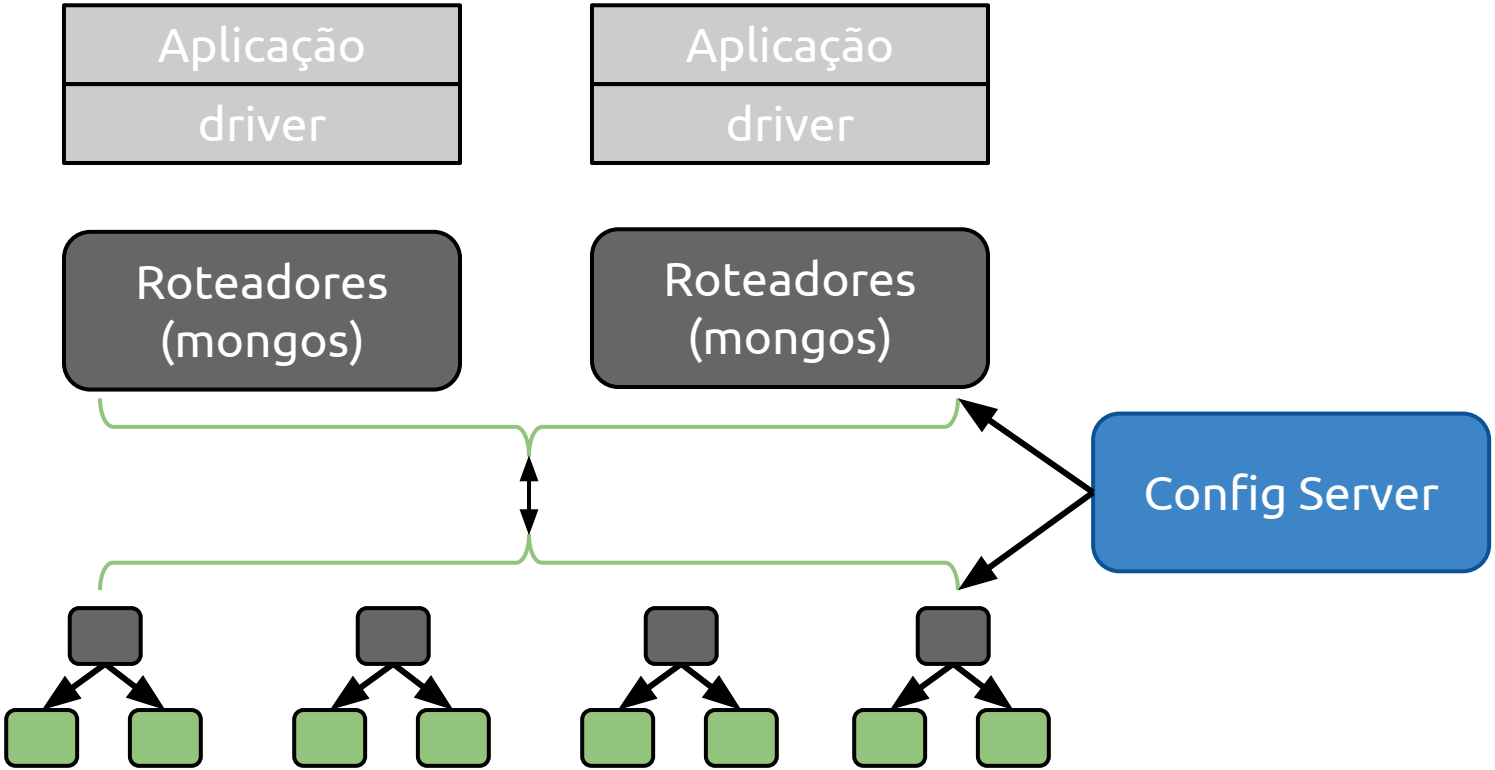
Sharding

- Escalabilidade
- Banco de dados é suportado em apenas uma máquina
- Dimunui o número de operações que um servidor apenas suporta
- Aumenta a complexidade de administração do banco

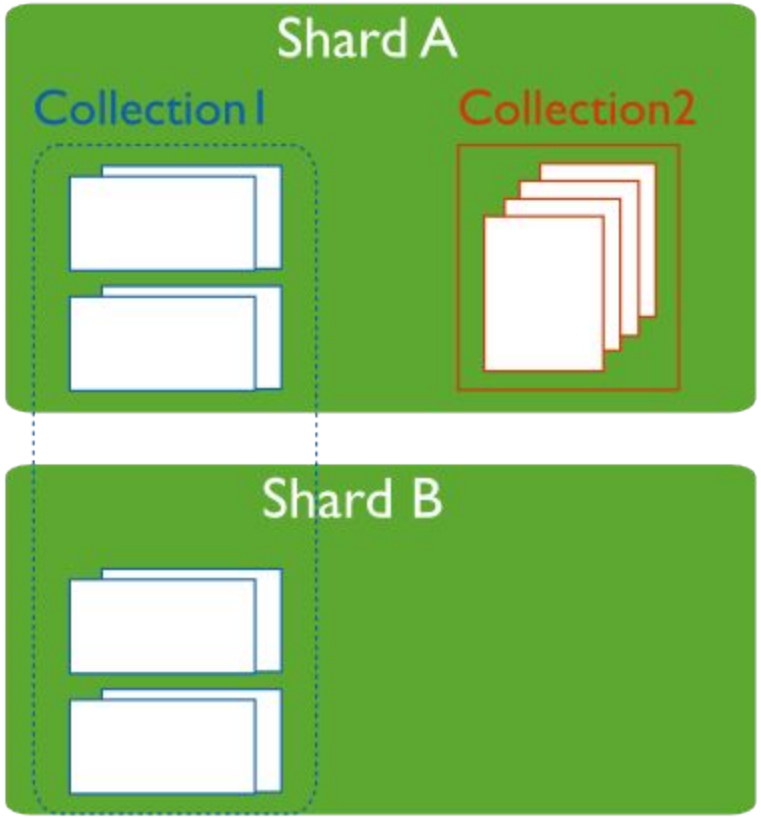
Sharding


Aplicação
driver
Aplicação
driver
Roteadores (mongos)
Roteadores (mongos)
Config Server
Shard (replica)
Shard (replica)

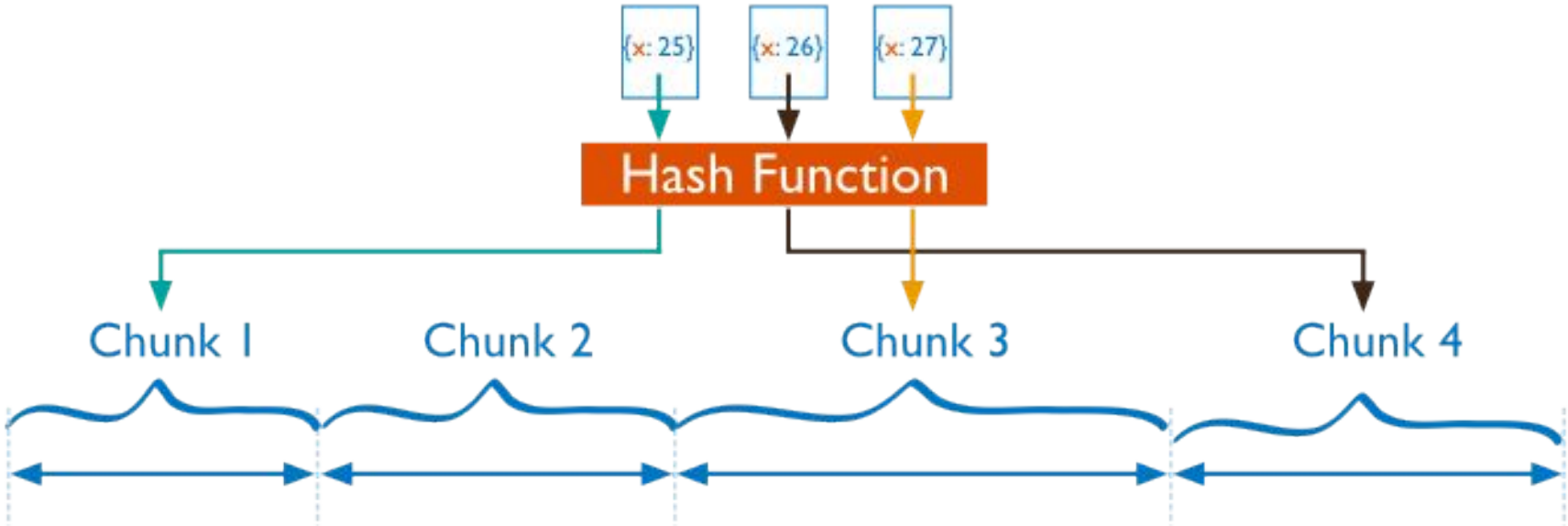
Sharding


Aplicação
driver
Aplicação
driver
Roteadores (mongos)
Roteadores (mongos)
Config Server

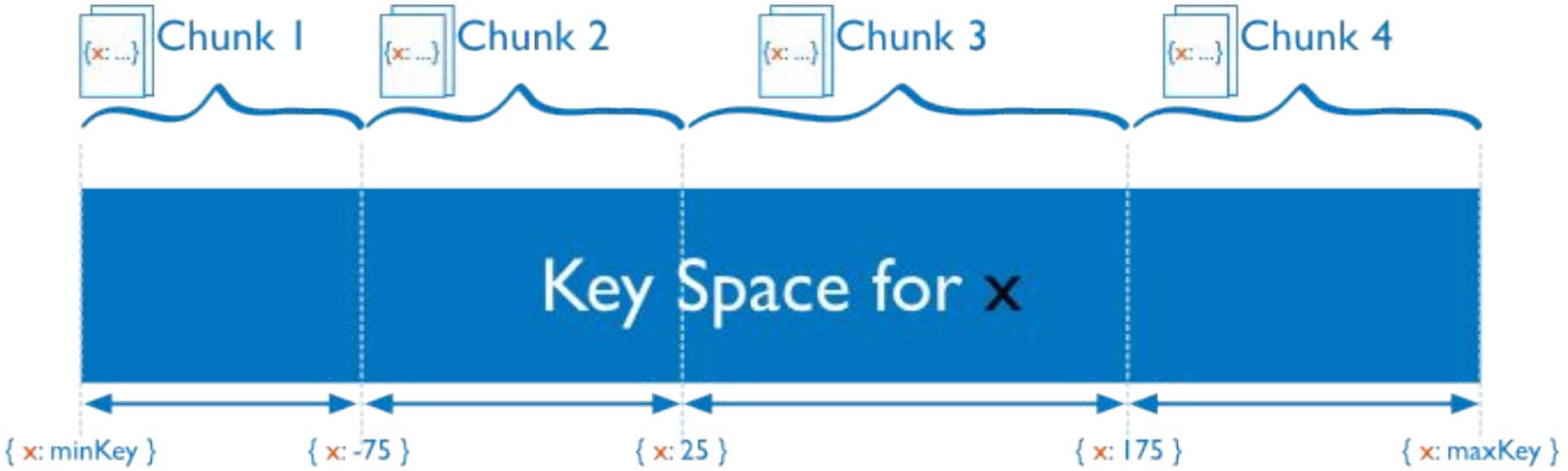
Sharding



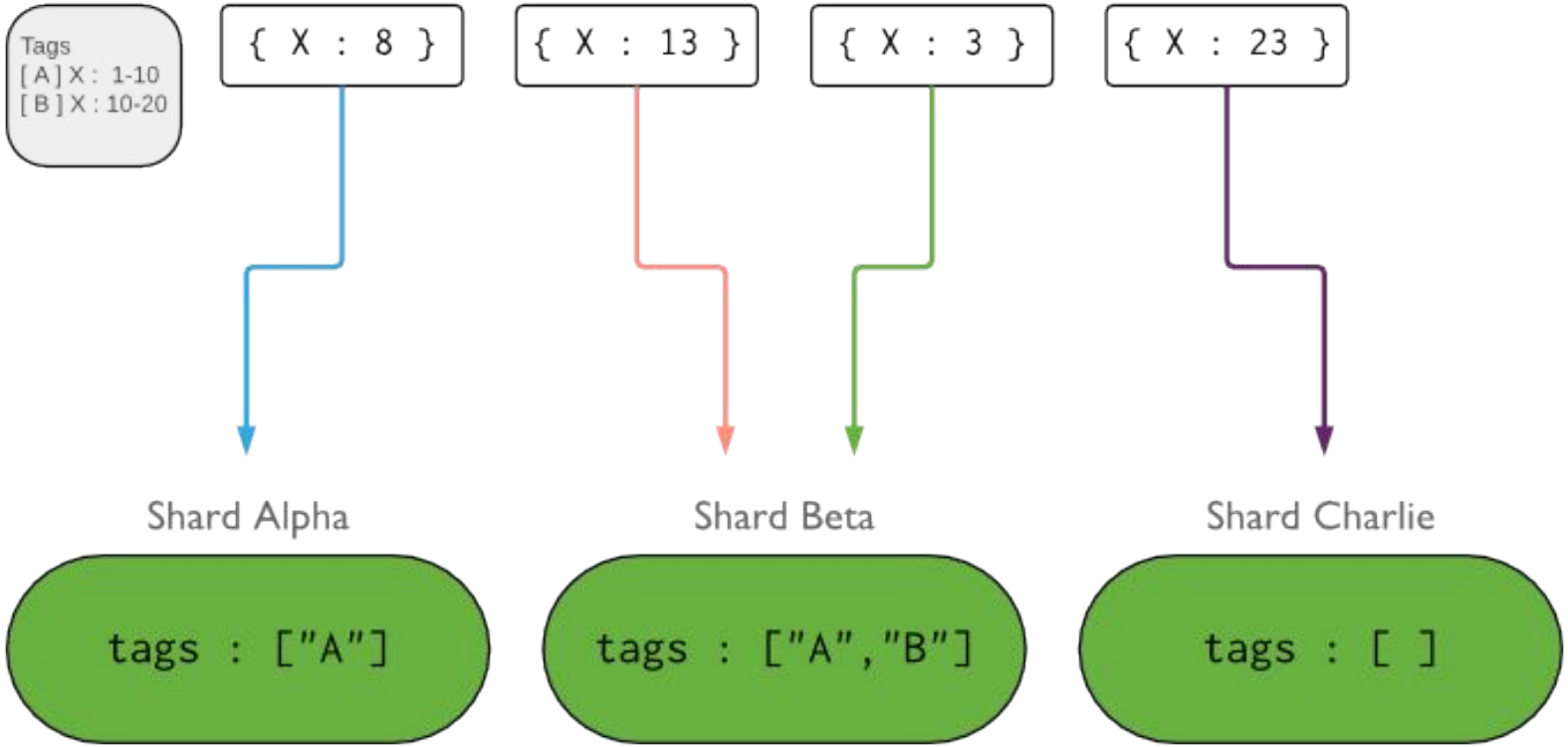
Sharding



Sharding


{x: ...}
Chunk 1
{x: ...}
Chunk 2
{x: ...}
Chunk 3
{x: ...}
Chunk 4
Key Space for x
{ x: minKey }
{ x: -75 }
{ x: 25 }
{ x: 175 }
{ x: maxKey }

Sharding


Tags
[A] X : 1-10
[B] X : 10-20
{ X : 8 }
{ X : 13 }
{ X : 3 }
{ X : 23 }
Shard Alpha
Shard Beta
Shard Charlie
tags : ["A"]
tags : ["A","B"]
tags : [ ]

Exemplo prático

- Criando replicas set
- Criando shards
- Inserindo dados no mongo shardeado

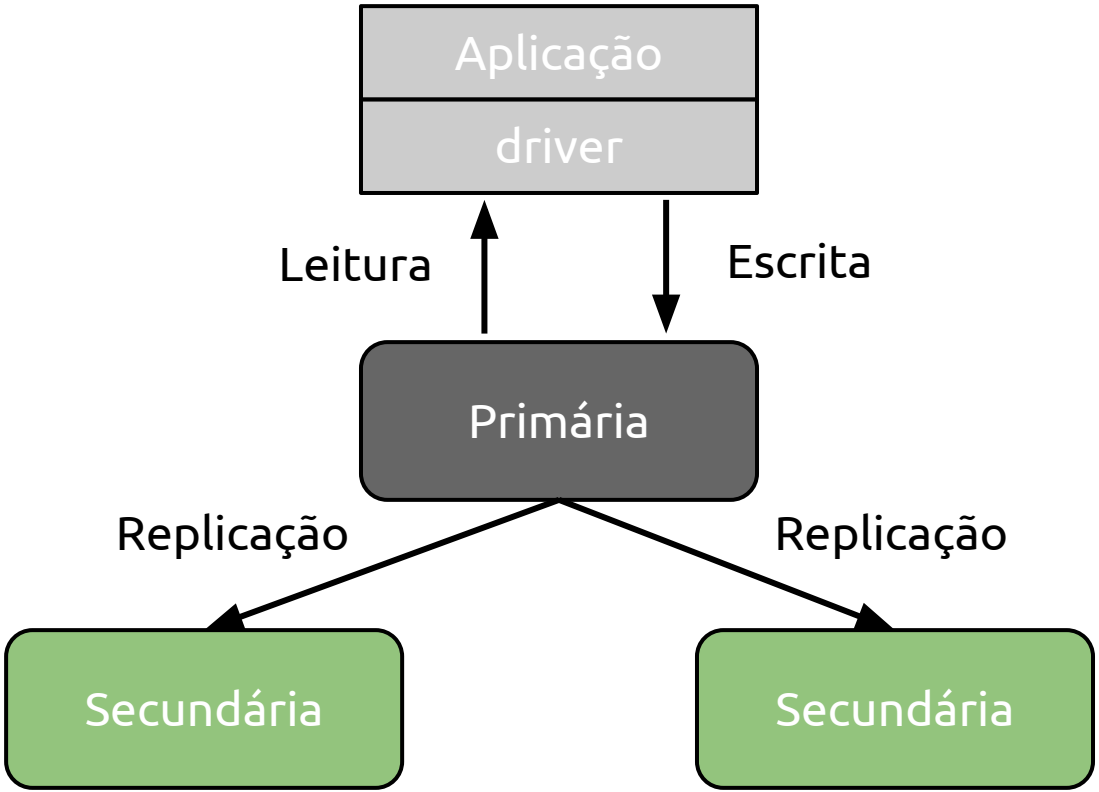
Backup

Simples

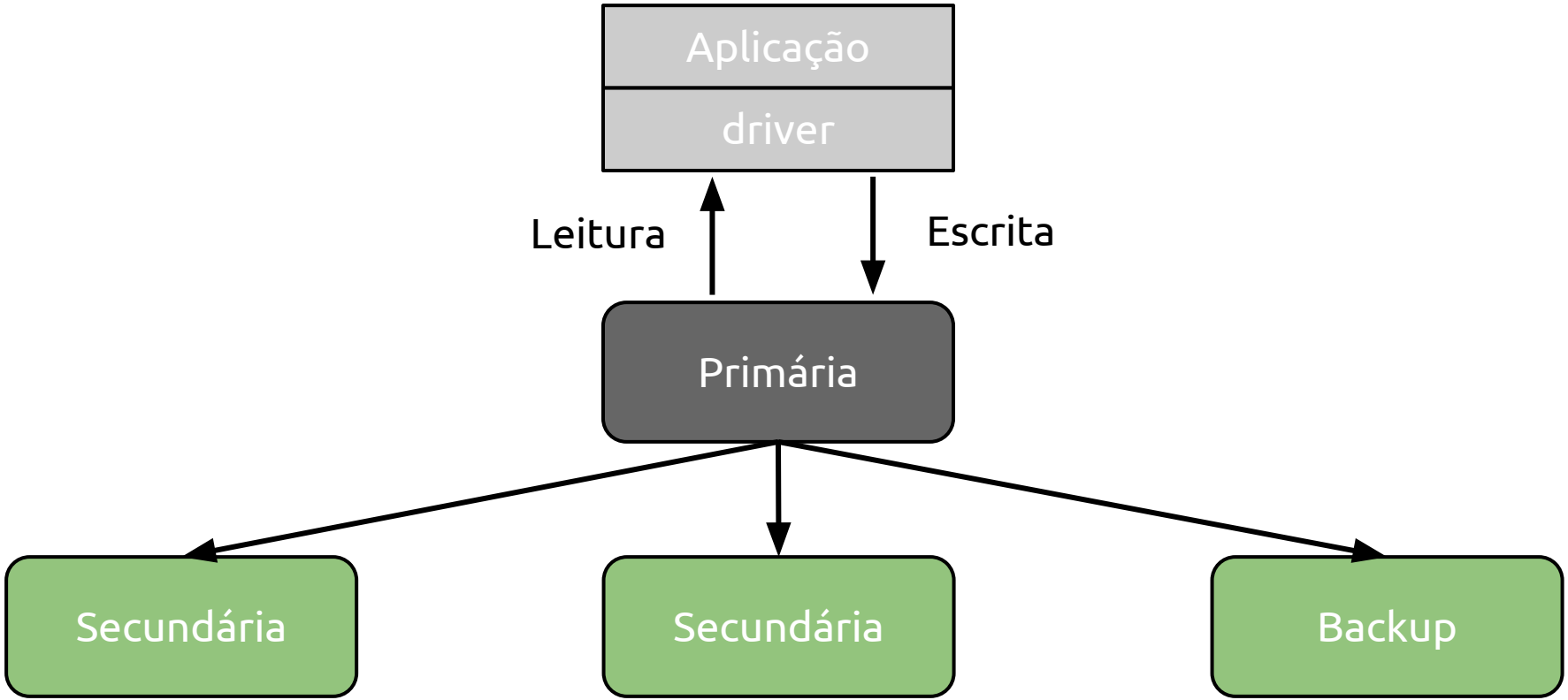
mongodump

- Funciona bem para uma estrutura simples
- Uma cópia da pasta de dados funcionaria da mesma maneira

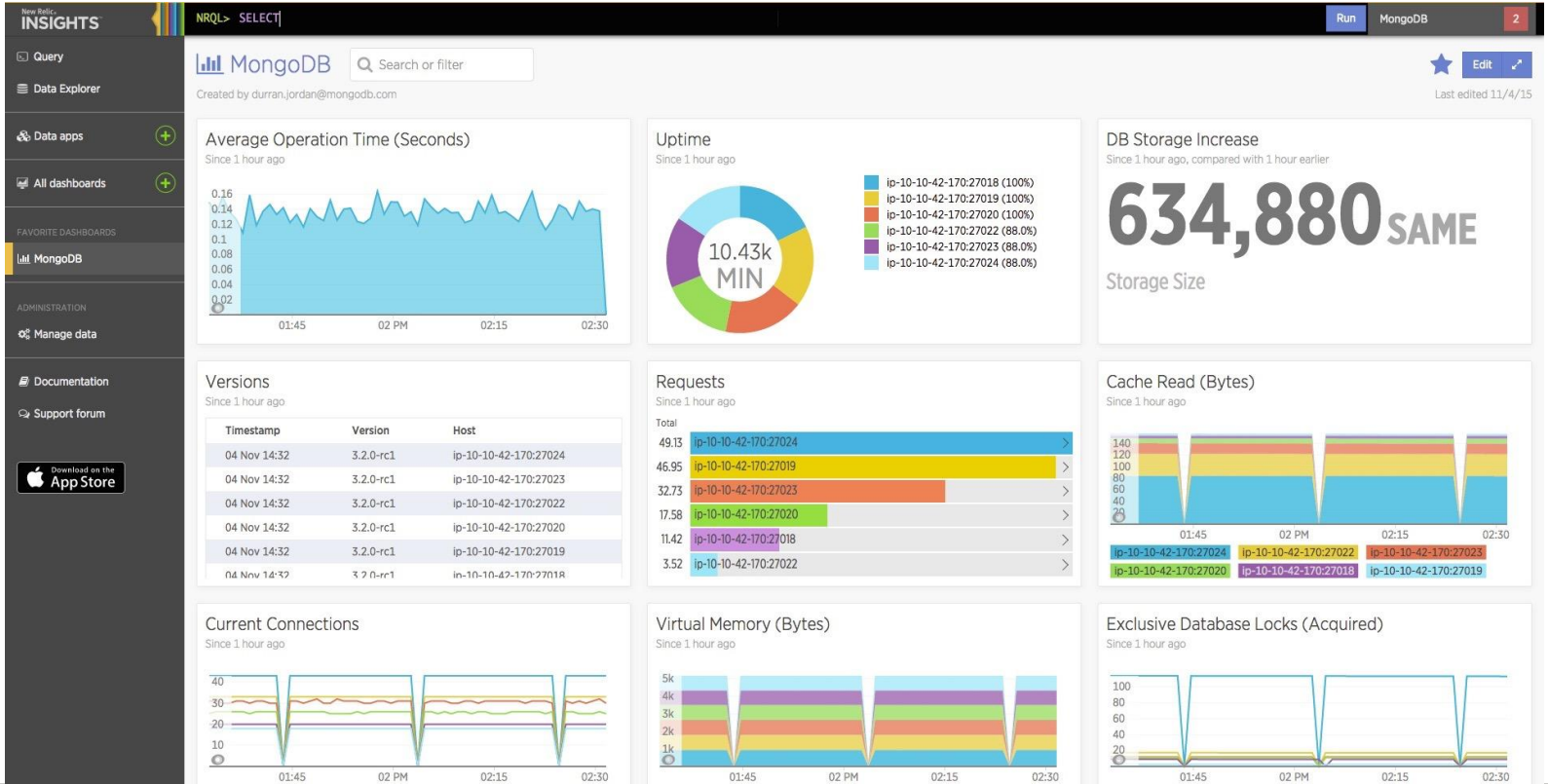
Replica set


Aplicação
driver
Leitura
Escrita
Primária
Replicação
Replicação
Secundária
Secundária

Replica set


Aplicação
driver
Leitura
Escrita
Primária
Secundária
Secundária
Backup

MongoDB Cloud Manager



Diferenças dos bancos relacionais?

- Tipos de índices que podem ser utilizados para arquivar vários tipos de dados diferentes;
- Operações de leitura e escrita são mais eficientes;
- Acesso agilizado por não utilizar joins, mas sim subdocumentos.
- Sharding

Preparando para os exercícios

Ligar a máquina virtual - NoSQL

1 - Abra **dois** terminais

2 - No primeiro terminal ligue o mongod com o comando:

```
cd ~/Aulas/mongodb; ./bin/mongod --dbpath=../dados
```

3 - No segundo terminal importe os dados para essa aula executando o comando:

```
cd ~/Aulas/mongodb; ./bin/mongoimport -d nosqlclass -c Vocabulary ../nosql-class/aula2/Vocabulary.json
```

4 - Após importar abra o mongo shell com o comando:

```
cd ~/Aulas/mongodb; ./bin/mongo
```

Exercício

Faça uma pesquisa simples na coleção Vocabulary pelo termo “feliz” no campo text e diga:

A) Número de documentos que foi escaneado

B) Tempo que levou para fazer a consulta

C) Crie um índice simples no campo text

D) Número de documentos que foi escaneado

E) Tempo que levou para fazer a consulta

Referências

http://crocodillon.com/blog/mongodb-for-dbas-introduction

http://waldyrfelix.com.br/2016/08/10/4-coisas-essenciais-sobre-mongodb/

https://www.youtube.com/watch?v=qI_g07C_Q5I

https://pt.wikipedia.org/wiki/%C3%81rvore_B